

Plan de cours N° : 582

Durée : 3 jours (21h)

PARTICIPANTS / PRE-REQUIS

Tous développeurs Java, et plus particulièrement ceux spécialisés dans le développement d'applications critiques, temps réel ou fonctionnant en environnement embarqués.

Connaissance des bases du langage Java.

OBJECTIFS PEDAGOGIQUES

Identifier les principes de fonctionnement d'une JVM. Développer et démarrer un agent JVMTI. Appliquer des "Guide Lines" de programmation pour optimiser les performances. Déterminer le fonctionnement du Garbage Collector et ses évolutions. Recycler les ressources de la JVM et optimiser les applications multithreadées. Utiliser des outils graphiques de monitoring et de profiling Java.

MOYENS PEDAGOGIQUES

Réflexion de groupe et apports théoriques du formateur

Travail d'échange avec les participants sous forme de réunion-discussion

Utilisation de cas concrets issus de l'expérience professionnelle

Validation des acquis par des exercices de synthèse

Alternance entre apports théoriques et exercices pratiques (en moyenne 30 et 70%)

Remise d'un support de cours.

Assistance post-formation d'une durée de 1 an sur le contenu de la formation via notre adresse mail dédiée formateurs@atp-formation.com

MOYENS PERMETTANT LE SUIVI DE L'EXECUTION ET DES RESULTATS

Feuille de présence signée en demi-journée,

Evaluation des acquis tout au long de la formation,

Questionnaire de satisfaction,

Attestation de stage à chaque apprenant,

Positionnement préalable oral ou écrit,

Evaluation formative tout au long de la formation,

Evaluation sommative faite par le formateur ou à l'aide des certifications disponibles.

MOYENS TECHNIQUES EN PRESENTIEL

Accueil des stagiaires dans une salle dédiée à la formation, équipée d'ordinateurs, d'un vidéo projecteur d'un tableau blanc.

MOYENS TECHNIQUES DES CLASSES A DISTANCE

A l'aide d'un logiciel comme Teams, Zoom etc... un micro et éventuellement une caméra pour l'apprenant, suivez une formation en temps réel et entièrement à distance. Lors de la classe en ligne, les apprenants interagissent et communiquent entre eux et avec le formateur.

Les formations en distanciel sont organisées en Inter-Entreprise comme en Intra-Entreprise. L'accès à l'environnement d'apprentissage (support de cours, labs) ainsi qu'aux preuves de suivi et d'assiduité (émargement, évaluation) est assuré.

Les participants recevront une convocation avec lien de connexion

Pour toute question avant et pendant le parcours, une assistance technique et pédagogique est à disposition par mail et par téléphone auprès de notre équipe par téléphone au 04.76.41.14.20 ou par mail à contact@atp-formation.com

ORGANISATION

Les cours ont lieu de 9h00-12h30 13h30-17h00.

PROFIL FORMATEUR

Nos formateurs sont des experts dans leurs domaines d'intervention

Leur expérience de terrain et leurs qualités pédagogiques constituent un gage de qualité.

ACCESSIBILITE

Les personnes atteintes de handicap souhaitant suivre cette formation sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités de suivre la formation.

MISE A JOUR

08/01/2024



Plan de cours N° : 582

Durée : 3 jours (21h)

Quelques rappels fondamentaux

- L'environnement J2SE (Java 2 Standard Edition)
 - Concepts fondamentaux (ClassLoader, JIT, GC, Gestionnaire d'exceptions...)
 - JVMPi (Java Virtual Machine Profiling Interface)
 - Quelques outils du J2SE (javap...)
- Notion de byte code
 - Utilisation du désassembleur javap
 - Notions de pile et de tas (Stack & Heap)
 - Mécanisme de déclarations : variables locales et attributs
 - Appels de méthodes en assembleur
- Quelques considérations sur les performances
 - Temps d'exécution VS taille de l'exécutable
 - Génie logiciel VS Performances

Développement d'agents JVMTI

- Concepts fondamentaux de JVMTI
 - Les différentes versions l'interface JVMTI
 - Les événements JVMTI
- Codage d'agents JVMTI
 - Traquer l'activité du ClassLoader
 - Traquer l'activité en termes d'allocations dynamiques
 - Traquer l'activité en termes d'appels de méthodes

"Guide lines" de programmation

- Utiliser correctement les Logger
 - Log4J...
- Fonctionnement de l'instruction switch
- Limitation d'instances temporaires
- Utiliser correctement les chaînes de caractères
- Utilisation des collections Java
 - Choisir les bons algorithmes face aux besoins
 - Collection synchronisées ou non (Vector vs ArrayList...)
 - Les différentes techniques de parcours d'une collection
 - Problématiques de l'autoboxing et du unboxing
- Utilisation de tableaux Java typés
 - Accès indexés aux valeurs
 - Traitements des valeurs de types primitifs
- Encapsulation VS performance

Le garbage collector (GC)

- Aspects fondamentaux
 - Tâches du garbage collector : libération et défragmentation
 - Les différents algorithmes utilisés
- Fonctionnement du Garbage Collector de la JVM Hotspot (Oracle)
 - Monitorer l'activité du GC
 - Collectes mineures et collectes majeures
- Comparaison Hotspot/OpenJ9
- Paramétrage du Garbage Collector de la JVM HotSpot
 - Gestion de la taille des Heap (-Xmx, -Xms, -XX:NewRatio, -XX:SurvivorRatio...)
 - Libération incrémentale des ressources
- Les évolutions du GC au fil des versions de Java

Recyclons les ressources de la JVM

- Utilisation d'un pool d'instances
 - Implémentation d'un pool d'instances
 - Utilisation d'un pool d'instances
 - Etude du comportement du GC via un agent JVMTI
- Applications multithreadées
 - Monitoring via JVMTI sur l'utilisation des threads
 - Le package `java.util.concurrent`
 - Utilisation de pool de threads
 - Gestion de la synchronisation de vos threads
- SoftReferences et WeakReferences

Outils graphiques de monitoring et de profiling

- La JConsole
 - Présentation du modèle JMX (Java Monitoring eXtensions)
 - Lancement et attachement de la Jconsole
 - Les différentes catégories d'informations collectées
- Visual GC et GC Viewer
 - Les différences entre les deux outils
 - Analyses poussées de l'activité du GC
- Java VisualVM
 - Présentation de l'outil
 - Profiling de type CPU
 - Profiling de type Memory
- Autres outils de profilage
 - Utilisation du plugin Eclipse Memory Analyser Tool